Computer Science

New Approximation Techniques for Some Ordering Problems

Satish Rao[1] *Andréa W. Richa*[2]

September 25, 1997

CMU-CS-97-177

Carnegie Mellon

DTIC QUALITY INSPECTED 3

New Approximation Techniques for Some Ordering Problems

Satish Rao[1] *Andréa W. Richa*[2]

September 25, 1997

CMU-CS-97-177

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

To appear in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, January 1998.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

19971105 044

[1]Research supported by NEC Research Institute, 4 Independence Way, Princeton NJ 08540. Email: satish@research.nj.nec.com.

[2]Research supported in part by: NSF National Young Investigator Award No. CCR-9457766, ARPA Contract F33615-93-1-1330, NEC Research Institute, and DIMACS. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. Email: aricha@cs.cmu.edu.

Keywords: Approximation algorithms, networks, graphs, ordering problems, linear arrangement.

Abstract

We describe $O(\log n)$ times optimal approximation algorithms for the NP-hard graph optimization problems of minimum linear arrangement, minimum containing interval graph, and minimum storage-time product. This improves on the $O(\log n \log\log n)$ approximation bounds provided in a previous paper by Even, Naor, Scheiber and Rao.

Our techniques are based on using spreading metrics (as in Even, Naor, Rao, and Scheiber) to define a cost estimate for a problem. In this paper, we develop a recursion where at each level we identify cost which, if incurred, yields a reduction in the spreading metric cost estimates for the resulting subproblems. Specifically, we give a strategy where the cost of solving a problem at a recursive level is C plus the cost of solving the subproblems, *and* where the spreading metric cost estimate on the subproblem(s) is reduced by at least $\Omega(C/\log n)$. This ensures that the resulting solution has cost at most $O(\log n)$ times the original spreading metric cost estimate.

We note that this is an existentially tight bound on the relationship between the spreading metric cost estimates and the true optimal values for these problems.

For planar graphs, we combine a structural theorem of Klein, Plotkin and Rao, with our new recursion and standard divide-and-conquer techniques to show that the spreading metric cost estimates are within an $O(\log\log n)$ factor of the cost of the optimal solution for the minimum linear arrangement and the minimum containing interval graph problems.

1 Introduction

We describe approximation algorithms that apply to the NP-hard graph optimization problems of minimum linear arrangement, minimum containing interval graph, and minimum storage-time product [3]. All of the ideas that we use for approximating the minimum containing interval graph, and for the minimum storage-time product problems can be illustrated by the algorithms for minimum linear arrangement. Thus, we restrict our exposition primarily to the minimum linear arrangement problem which we define as follows.

Let G be a graph with associated edge weights. Informally, a minimum linear arrangement of G is an embedding of G in the linear array such that: (i) we have a one-to-one mapping from the nodes of G to the nodes of the linear array, and (ii) the weighted sum of the edge lengths defined by the distance along the linear array between its endpoints, that is, the *cost* of the linear arrangement, is minimum.

Finding a minimum linear arrangement is NP-hard, even for the case when all edges have unit weight. We present a polynomial time $O(\log n)$ approximation algorithm for the minimum linear arrangement problem on a graph with n nodes. This improves the best previous approximation bound of Even, Naor, Rao, and Scheiber [1] for this problem by a $O(\log\log n)$ factor. If the graph is planar (or, more generally, if it excludes $K_{r,r}$ as a minor, for fixed r, where $K_{r,r}$ is the $r \times r$ complete bipartite graph), we obtain an $O(\log\log n)$ approximation factor for the minimum linear arrangement problem, using a variation of the algorithm presented for the general case. We obtain this improvement by combining the techniques used for the general case with the algorithm presented by Klein, Plotkin, and Rao [5] for finding separators in graphs that exclude fixed $K_{r,r}$ minors.

By using techniques from [8], we can view the interval graph containment problem (which we define formally later) as a vertex version of the minimum linear arrangement problem. Thus, we also obtain an $O(\log n)$ approximation algorithm for general graphs and an $O(\log\log n)$ approximation algorithm for planar graphs for this problem. This improves on previous bounds of $O(\log n \log\log n)$ [1] for general graphs and $O(\log n)$ for planar graphs.

Also, using ideas from [8], we can extend our ideas to produce an $O(\log n)$ approximation for the minimum storage-time product problem, improving on a previous bound of $O(\log n \log\log n)$ in [1].

1.1 Previous Work

Leighton and Rao [6] presented an $O(\log n)$ approximation algorithm for balanced partitions of graphs. Among other applications, this provided $O(\log^2 n)$ approximation algorithms for the minimum feedback arc set and for the minimum-cut linear arrangement problem. Hansen [4] used the ideas in [6] to present $O(\log^2 n)$ approximation algorithms for the minimum linear arrangement problem, and for the more general problem of graph embeddings in d-dimensional meshes. Ravi et al. [8] presented polynomial time $O(\log^2 n)$ approximation algorithms for the minimum storage-time product problem where all tasks have the same execution time, and for the minimum containing interval graph. The minimum storage-time product problem also generalizes the minimum linear arrangement problem.

Seymour [9] was the first to present a directed graph decomposition divide-and-conquer approach that does not rely on balanced cuts. He presented a polynomial time $O(\log n \log\log n)$ approximation algorithm for the minimum feedback arc set problem. Even, Naor, Rao, and Scheiber [1] extended the spreading metric approach used by Seymour to obtain a polynomial time $O(\log n \log\log n)$ approximation algorithm for the minimum linear arrangement problem, as well as for the minimum storage-time product and for the minimum containing interval graph

problems. Even et al. actually show similar approximation results for a broader class of problems, namely the ones that satisfy their divide-and-conquer paradigm: "A graph optimization problem where a divide-and-conquer approach is applicable and where a fractional spreading metric is computable in polynomial time" satisfies this paradigm. They present a polynomial time $O(\min\{\log W \log\log W, \log k \log\log k\})$ approximation algorithm for these problems, where k denotes the number of "interesting" nodes in the problem instance (clearly $k \leq n$), and W is the lower bound on the cost of solving the optimization problem provided by the spreading metric. Examples of such problems, besides the ones already mentioned, are: graph embeddings in d-dimensional meshes, symmetric multicuts in directed networks, multiway separators and ρ-separators (for small values of ρ) in directed graphs. For a detailed description of each of those problems, see [1].

Even, Naor, Rao, and Scheiber [2] extend the spreading metric techniques to graph partitioning problems. They use simpler recursions that yield a logarithmic approximation factor for balanced cuts and multiway separators. However, they were not able to extend this simpler technique to obtain a logarithmic approximation bound for the problems considered in [1].

1.2 Spreading metrics and consumptive recursions

Our algorithms use an approach that relies on *spreading metrics.* Spreading metrics have been used in recent divide-and-conquer techniques to obtain improved approximation algorithms for several graph optimization problems that are NP-hard [1]. These techniques perform the divide step according to the *cost* of solving the subproblems generated, rather than according to the *size* of such subproblems.

A spreading metric on a graph is an assignment of lengths to the edges of the graph that has the property of "spreading apart" (with respect to the metric lengths) the nontrivial subgraphs. The *volume* of the spreading metric is the sum, taken over all edges, of the length of each edge multiplied by its weight. The volume of a spreading metric provides a lower bound on the cost of solving the optimization problem in the input graph.

Our techniques are based on using spreading metrics (as in [1]) to define a cost estimate for a problem. In this paper, we develop a recursion where at each level we identify cost which, if incurred, yields a reduction in the spreading metric cost estimates for the resulting subproblems. Specifically, we give a strategy where the cost of solving a problem at a recursive level is C plus the cost of solving the subproblems, *and* where the spreading metric cost estimate on the subproblem(s) is reduced by at least $\Omega(C/\log n)$. This ensures that the resulting solution has cost at most $O(\log n)$ times the original spreading metric cost estimate.

The recursion is based on divide-and-conquer; that is, find an edge set whose removal divides the graphs into subgraphs, and then recursively order the subgraphs. The cost of this recursive level being the cost associated with the edges in the cut. Previous recursive methods and analyses proceeded by finding a small cutset of edges where the maximum spreading metric volumes of the subproblems were quickly reduced. We proceed by finding a *sequence* of cutsets whose total cost can be upper bounded, say by a quantity C, and whose total spreading metric volume is at least $\Omega(C/\log n)$ as stated above. The crux of the argument is that, for the linear arrangement problem, the cost of an edge in a cutset can be bounded by the number of nodes between the previous and the next cutset in the sequence.

We point out that the methods in [1] applied to more problems including d-dimensional graph embedding and feedback arc set [9]. We could not extend our methods to these problems since we were unable to find a suitable bound on the cost of a sequence of cutsets.

Finally, for planar graphs and other constant minor excluded graphs, we combine a structural theorem of Klein, Plotkin and Rao [5], with our new recursion and standard divide-and-conquer

techniques to show that the spreading metric cost estimates are within an $O(\log\log n)$ factor of the cost of the optimal solution for the minimum linear arrangement and the minimum containing interval graph problems.

1.3 Overview

We present a formal definition of the problems, and of the spreading metric used in Section 2 and Section 3, respectively. In Section 4, we present a polynomial time $O(\log n)$ approximation algorithm for the minimum linear arrangement problem on an arbitrary graph with n nodes and nonnegative edge weights $w(e)$. In Section 5, we show how to improve the approximation factor to $O(\log\log n)$ in case the graph has no fixed $K_{r,r}$ minors, e.g., the graph is planar.

2 The Problems

In this section, we present definitions for the ordering problems considered in this paper.

2.1 Minimum Linear Arrangement

The *minimum linear arrangement (MLA) problem* is defined as follows: Given an undirected graph (network) $G(V,E)$, with $n = |V|$ and $m = |E|$, and nonnegative edge weights $w(e)$, for all e in E, we would like to find a linear arrangement of the nodes $\sigma : V \rightarrow \{1,\ldots,n\}$ that minimizes the sum, over all (i,j) in E, of the weighted edge lengths $|\sigma(i) - \sigma(j)|$. In other words, we would like to minimize the *cost* $\sum_{(i,j)\in E} w((i,j))\,|\sigma(i) - \sigma(j)|$. In the context of VLSI layout, $|\sigma(i) - \sigma(j)|$ represents the length of the interconnection between i and j.

2.2 Minimum Storage-Time Product

The minimum storage-time product problem arises in a manufacturing or computational process, in which the goal is to minimize the storage-time product of the process: We want to minimize the use of storage over time, assuming storage is an expensive resource. This problem generalizes the minimum linear arrangement problem. It is also a generalization of the single-processor scheduling problem, if we are minimizing the weighted sum of completion times (this problem is NP-complete [3, problem SS13, page 240].).

Let $G(V,E)$ be a directed acyclic graph with edge weights $w(e)$, for all e in E. The nodes of G represent tasks to be scheduled on a single processor. The time required to execute task v is given by $p(v)$. The weight on edge (u,v), $w(u,v)$, represent the number of units of storage required to save the intermediate results generated by task u until they are consumed at task v. The *minimum storage-time product problem* consists of finding a topological ordering of the nodes $\sigma : V \rightarrow \{1,\ldots,n\}$ that minimizes $\sum_{(i,j)\in E,\sigma(i)<\sigma(j)}\{w((i,j))[\sum_{k\,:\,\sigma(i)<\sigma(k)<\sigma(j)} p(\sigma(k))\,]\}$.

2.3 Minimum Containing Interval Graph

An *interval graph* is a graph whose vertices can be mapped to distinct intervals in the real line such that two vertices in the graph have an edge between them iff their corresponding intervals overlap. An interval graph completion of a graph G consists of an interval graph with same node set as G that contains G as a subgraph.

We use the following characterization of interval graphs, due to [7]. An undirected graph $G(V,E)$ is an interval graph iff there exists a linear ordering $\sigma : V \rightarrow \{1,\ldots,n\}$ of the nodes in V such that: If a node u has an edge to node v, where $\sigma(u) < \sigma(v)$, then every node w such that $\sigma(u) < \sigma(w) < \sigma(v)$, also has an edge to node v. Note that such an ordering of the vertices of G uniquely defines a completion of G into an interval graph. The cost of such a completed graph is given by the total number of edges in the graph. The *minimum containing interval graph problem* consists of computing a linear ordering σ that gives an interval graph that is a completion of G

and that has minimum cost.

This problem arises, e.g., in archeology when finding a consistent chronological model for tool use while making as few assumptions as possible.

3 Spreading Metric

In this section we define the spreading metric used in the algorithms for the MLA problem presented in Sections 4 and 5. Analogous functions are used when approximating the minimum storage-time product problem and the minimum containing interval graph problem (see [1]).

A *spreading metric* is a function $\ell : E \to \mathbf{Q}$ that assigns rational lengths to every edge in E, and that can be computed in polynomial time. It also needs to satisfy the following two properties (here we present the spreading metric properties in the context of the MLA problem; see [1] for a more general definition):

1. *Lower bound:* The *volume* of the spreading metric, defined by $\sum_{e \in E} w(e)\ell(e)$, is a lower bound on the cost of a MLA of G.

2. *Diameter guarantee:* Let the distances be measured with respect to the lengths $\ell(e)$. The distances induced by the spreading metric "spread" the graph and all its nontrivial subgraphs. In this application (as in all applications that appear in [1]) this translates to: The diameter of every nontrivial connected subgraph U of V is $\Omega(|U|)$.

Let $\mathcal{V}$ denote the set of all nontrivial connected subgraphs of V. The spreading metric ℓ used is a solution of the linear program below. That is, ℓ minimizes $\sum_{e \in E} w(e)\ell(e)$, while satisfying the constraints that follow:

$$W = \min \sum_{e \in E} w(e)\ell(e) \tag{1}$$

$$\text{s.t.} \quad \Big(\sum_{u \in U} \mathsf{dist}(u, v)\Big)/|U| > |U|/4, \forall U \in \mathcal{V}, \forall v \in U \tag{2}$$

$$\ell(e) \geq 0, \forall e \in E \tag{3}$$

where $\mathsf{dist}(u, v)$ is the length of a shortest path from u to v according to the lengths $\ell(e)$; W is the volume of the spreading metric ℓ. See [1] for proofs that a solution to (1-3) satisfies the spreading metric properties, and that such a solution can be computed in polynomial time.

In the remaining of this paper all the distances in G are measured with respect to σ.

4 The algorithm

Let $G(V, E)$ be a graph with nonnegative edge weights $w(e)$. Assume w.l.o.g. that G is connected (otherwise consider each connected component of G separately) and that all the edge weights $w(e)$ are at least 1. Let ℓ be a spreading metric obtained by solving (1-3) for G, and let W be the volume of this spreading metric. Fix a node v in V. An edge (x, y) is at *level* i with respect to v iff $\mathsf{dist}(v, x)$ is at most i and $\mathsf{dist}(v, y)$ is greater than i. Let the *weight of level* i, denoted by ρ_i, be the sum of the weights of the edges at level i (notice that an edge may be at more than one level). Let $\alpha_k = (W2^k)/n$, for $-\log(W/n) \leq k \leq \log n$. (W.l.o.g. assume that $\log(W/n)$ and $\log n$ are integers.) Level i has *index* k, $-\log(W/n) \leq k \leq \log n$, iff ρ_i belongs to the interval $I_k = (\alpha_{k-1}, \alpha_k]$.

It follows from Equation (3) that we must have at least $n/4$ distinct levels with nonzero weight. Thus, there must be at least $n/(4 \log W)$ levels with same index k, for some k. Let κ be the exact number of levels of index k.

We cut along the κ levels of index k (i.e., we remove the edges that are at those levels, even if they are also at some other level of index different than k). For all i, let level a_i be the the ith level, in increasing order of the levels, of index k. Let H_i be the subgraph induced by the nodes that are at distance greater than a_i and at most a_{i+1} from v. Let H_0 (resp., H_κ) be the subgraph induced by the nodes that are at distance at most a_1 (resp., greater than a_κ) from v. Let n_i denote the number of nodes at H_i. We recurse on each H_i, obtaining a linear arrangement σ_i for the n_i nodes in this subgraph. We combine the linear arrangements obtained for the H_i's, obtaining a linear arrangement σ for G, as follows:

$$(\sigma(1), \ldots, \sigma(n)) = (\sigma_0(1), \ldots, \sigma_0(n_1), \sigma_1(1), \ldots, \sigma_1(n_2),\ \ldots,\ \sigma_\kappa(1), \ldots, \sigma_\kappa(n_\kappa))$$

This algorithm runs in polynomial time.

Let $C(Z)$ be the cost of the linear arrangement obtained by our algorithm for a graph whose spreading metric volume is Z. Then:

$$C(W) \leq C(W - \sum_{i=1}^{\kappa} \rho_{a_i}) + \sum_{i=1}^{\kappa} [\rho_{a_i}(n_{i-1} + n_i)] \tag{4}$$

If some edge e in the ith level also belong to some other level of index k, say the jth such level, then this edge also belongs to every level of index k between the ith and jth levels. W.l.o.g. assume $i < j$. Informally, edge e will be "stretched over" all the nodes in $H_i \cup \ldots \cup H_{j-1}$ in the linear arrangement produced by our algorithm. We "charge" for the portion of the edge that is stretched over the nodes in H_p and H_{p-1}, when considering the level of index k between these subgraphs, for all $i \leq p \leq j$.

We now show that $C(W) = O(W \log n)$. Since W is a lower bound on the cost of a linear arrangement of G, the cost of the linear arrangement obtained is at most $O(\log n)$ times the cost of a MLA. We first prove the following theorem:

Lemma 4.1 $C(W) \leq cW \log W$, *for some constant* c.

Proof: We will use induction on W. The base case $W = 0$ corresponds to a totally disconnected graph (a graph with no edges), and therefore $C(W) = 0$.

Combining Equation 4 with the fact that $\alpha_{k-1} < \rho_{a_i} \leq \alpha_k$, for all i, we obtain:

$$\begin{aligned}
C(W) &\leq C(W - \frac{\alpha_{k-1} n}{4 \log W}) + \alpha_k \sum_i (n_{i-1} + n_i) \\
&\leq c[W - \frac{\alpha_{k-1} n}{4 \log W}] \log[W - \frac{\alpha_{k-1} n}{4 \log W}] + 2\alpha_k n \\
&\leq c[W - \frac{\alpha_{k-1} n}{4 \log W}] \log W + 2\alpha_k n \\
&\leq cW \log W + \alpha_k n[2 - \frac{c}{8 \log W} \log W] \\
&\leq cW \log W
\end{aligned}$$

for a sufficiently large constant c. The second inequality follows from the induction hypothesis; the third inequality follows since $\alpha_k = 2\alpha_{k-1}$.

■

We still need to show how to bring the approximation factor down from $O(\log W)$ to $O(\log n)$. Consider the set E' of edges e such that $w(e) \leq W/(mn)$. Since any edge has length at most n in

any linear arrangement for G, their contribution to the solution of the MLA is at most W. Thus if we delete those edges and apply a ρ-approximation algorithm to the resulting graph, we obtain a linear arrangement of G, by simply adding those edges back into the linear arrangement found, with cost that is within a $(\rho + 1)$ factor of the cost of a MLA of G.

We now round down each weight $w(e)$, for all e in $E \backslash E'$, to its nearest multiple of $W/(mn)$. The error incurred by this rounding procedure is again at most W. Furthermore, we scale the rounded weights by $W/(mn)$, obtaining new weights for the edges that are all integers in the interval $[0, mn]$. Note that we have only changed the units in which the weights are expressed. Hence we obtain a polynomial time $(\rho + 2)$-approximation algorithm for the MLA problem on G with weights $w(e)$, by solving the MLA problem on $G \setminus E'$ with integral weights that belong to $[0, mn]$. The volume W' of the spreading metric for the latter problem is at most a polynomial in n. By Lemma 4.1, we have $C(W') \leq cW' \log(W') = c'W' \log n$, for some constant c'. Rescaling the edge weights back by multiplying $C(W')$ by $W/(mn)$, we obtain a minimum linear arrangement for the original weights on $G \setminus E'$ with cost at most $c'W \log n$.

5 Graphs with Excluded Minors

In this section we show how to obtain, in polynomial time, an $O(\log \log n)$ approximation bound for the MLA problem on a graph G with no fixed $K_{r,r}$ minors, e.g., a planar graph G. We denote by $K_{r,r}$ the $r \times r$ complete bipartite graph.

Definition 5.1 *Let H and G be graphs. Suppose that (i) G contains disjoint connected subgraphs A_v, for each node v of H, and (ii) for every edge (u, v) in H, there is a path $P_{(u,v)}$ in G with endpoints in A_u and A_v, such that any node in $P_{(u,v)}$ other than its endpoints does not belong to any A_w, w in H. Then $\cup_v A_v$ is said to be an* H-minor *of G.*

Klein et al. [5], show how to decompose (in polynomial time) a graph with no $K_{r,r}$-minor into connected components of small diameter. For our application, this implies that each connected component has at most a constant fraction of the nodes in G.

5.1 The Algorithm

We recursively solve the problem, as we do in the general case. We combine the partial solutions returned by each recursive step, as well as charge for each edge removed at a cut step, in the same way as in Section 4. It is in the way we decompose the graph before a recursive step that the algorithm of Section 4 differs considerably from the one presented in this section. Before each recursive step of this algorithm, we will perform a series of *shortest path levelings* on each induced connected subgraph, until we can guarantee that the original graph has been decomposed into subgraphs that contain at most a constant fraction of the nodes each, as follows. In the algorithm of Section 4, we perform only one shortest path leveling before each recursive step.

Let $G(V, E)$ be a graph that excludes $K_{r,r}$ as a minor, for some fixed $r > 0$. Let W be the volume of the spreading metric ℓ obtained when solving (1-3) for G. The algorithm proceeds in rounds. In each round we have a *cut step*, which corresponds to the series of cuts performed during the round, and a *recursive step*, which consists of recursing on the connected components that result from the cut step. Let $G_0, \ldots G_t$, $t \leq r$, be the series of subgraphs of G that we obtain during a round of the algorithm. At the beginning of the round for graph G (with n nodes) we let $G_0 = G$ and $s = n/b$, where b is a constant to be specified later. Let $i = 0$.

Let $n(G_i)$ denote the number of nodes in G_i. Fix a node v in G_i. A *shortest path leveling (SPL)* of G_i rooted at v consists of assigning levels to the edges of G_i as follows: An edge (x, y) is at *level*

i of this SPL iff $\mathsf{dist}(v, x)$ is at most i and $\mathsf{dist}(v, y)$ is greater than i. (An edge may be at more than one level.)

Group the levels of this SPL into "bands" of $2s$ consecutive levels. Alternate coloring the bands blue and red, in increasing order of the levels. W.l.o.g., assume that the subgraph induced by the blue bands has at least $n(G_i)/2$ nodes. We will now consider $2s$ cuts of the following type: For $0 \le j \le 2s-1$, a *leveled cut* j consists of all the edges in the jth level (with respect to distance from v) of every red band. That is, if the band consisting of the first $2s$ levels is colored blue, then the leveled cut j consists of the levels $2s+j, 6s+j, \ldots$, for all j. The spreading metric diameter guarantee implies that this SPL has at least $n(G_i)/4 = \Omega(n)$ levels.

Let $\beta_k = W2^k/(s \log n)$, for $0 < k < 2 \log\log n$. Let $\beta_0 = 0$ and $\beta_{(2\log\log n)} = W$. The *weight* of leveled cut j is the sum of the weights of the levels in the cut (the weight of an level being the sum of the weights of the edges at that level). Leveled cut j has *index* k, $0 \le k \le 2\log\log n - 1$, iff the weight of cut j belongs to the interval $I_k = (\beta_k, \beta_{k+1}]$. There must exist at least $2s/(2\log\log n)$ leveled cuts with same index k_i (assume $b \ge 8$). Thus k_i is at most $\log\log n + \log\log\log n < 2\log\log n - 1$, for n sufficiently large.

If $k_i > 0$, then we cut along these at least $s/(\log\log n)$ leveled cuts of index k_i, and recurse on the resulting connected components. In this case, $t = i$. Otherwise, we first cut along only one of the leveled cuts of index $k_i = 0$ (chosen arbitrarily). Then we check whether there exists a connected component G_{i+1} of G_i with more than $n(G_i)/2$ nodes. In case no such component exists, then $t = i$ and we recurse. Otherwise, we further proceed in performing a SPL on G_{i+1}, following the procedure just described, taking $i = i+1$.

Suppose we performed a series of r SPL's and corresponding cut procedures. The last cut performed, on G_{r-1}, generates a collection of connected components of this subgraph. Klein et al. [5] proved that the distance in the graph G between any pair of nodes in any such component is at most $n/6$, for a suitably chosen constant b, e.g., $b = \max(8, 6r^2)$.

Fix any node u in G. It follows from (2), that any subgraph of G with $(n-x)$ nodes that contains u has a node at distance at least $(n-x)/4$ from u. Suppose we start with the graph G. We proceed by removing one node at a time, choosing always to remove a node that has maximum distance to u, among the remaining nodes. Thus, we need to remove at least a $1/3$ fraction of the nodes before we are left with nodes that are within distance $n/6$ from u in G. Hence any resulting connected component of G_{r-1} has at most $2n/3$ nodes. Any other resulting connected component, in $G \setminus G_{r-1}$, has at most $n/2$ nodes, by the choice of the G_i's.

We distinguish between two types of cut steps: if k_t is equal to 0, then we have a cut step of *type I* in this round; otherwise k_t is not 0, and the cut step in this round is of *type II*. Note that $k_t = 0$ implies $k_j = 0$, for all $j \le t$. Let $C(Z, x)$ denote the cost of the linear arrangement obtained by our algorithm for a graph with x nodes and spreading metric volume Z.

Lemma 5.1 *$C(W, n) \le cW \log\log W$, for some constant c.*

Proof: The base cases for $W = 0$ or $n = 0$ are trivial. Suppose we perform a cut step of type I. Let the connected components resulting from this step be $H_0, \ldots, H_p$. Then:

$$\begin{aligned} C(W,n) &\le \sum_{i=0}^{p} C(W_i, n_i) + r\frac{2W}{s\log n} n \\ &\le \sum_i cW_i \log\log(2n/3) + \frac{2brW}{\log n} \\ &\le cW\log\log n - \frac{cW}{3\log n} + \frac{2brW}{\log n} \end{aligned}$$

$$\leq cW \log\log W$$

where W_i and n_i are the volume and number of nodes, respectively, associated with component H_i. We have shown that every n_i is at most $2n/3$. The second inequality above follows by induction. Note that $\log\log(2n/3) \leq \log\log n - 1/(3\log n)$. Thus the third inequality follows. The last inequality above follows for a sufficiently large constant c.

If the cut step performed was of type II, then we performed a series of $t \leq r$ SPL's and respective cut procedures. The last term on the rhs of the first inequality below accounts for the first $(t-1)$th leveled cuts used. The second term on the rhs of that inequality accounts for the tth leveled cut used. The charging scheme for the edges removed in the leveled cut of round t is analogous to the one used in Section 4.

$$\begin{aligned}
C(W,n) &\leq C(W - \frac{\beta_k s}{\log\log n}, n) + +2\beta_{k+1}n + (r-1)\frac{2W}{s\log n}n \\
&\leq C(W - \frac{\beta_k s}{\log\log n}, n) + (r+3)\beta_k n \\
&\leq c(W - \frac{\beta_k s}{\log\log n})\log\log W + (r+3)\beta_k n \\
&\leq cW\log\log W + \beta_k n(r + 3 - \frac{c\log\log W}{b\log\log n}) \\
&\leq cW\log\log W
\end{aligned}$$

The second inequality above follows from $\beta_k \geq 2W/(s\log n)$, and from $\beta_{k+1} = 2\beta_k$, $0 < k < 2\log\log n - 1$.

■

We use the same approach as described in Section 4 to bring the approximation factor down from $\log\log W$ to $O(\log\log n)$.

6 Conclusion

We provided an existentially tight bound on the relationship between the spreading metric cost estimates and the true optimal values for the problems of minimum linear arrangement, minimum containing interval graph, and minimum storage-time product.

It would be interesting to extend our techniques to obtain $O(\log n)$ approximation factors to other problems. In particular, it seems natural to extend our techniques to improve the best known approximation factors for other problems that satisfy the "approximation paradigm" of [1]. We would then also provide an existentially tight bound, on the ratio between the value of an optimal solution and the spreading metric cost estimate, to these problems.

However, since the approach used here depends on the structure of graph ordering problems, it remains an open question whether such extensions are possible.

References

[1] G. Even, J. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 62–71, October 1995.

[2] G. Even, J. Naor, S. Rao, and B. Schieber. Spreading metric based approximate graph partitioning algorithms. manuscript, Nov. 1995.

[3] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, NY, 1979.

[4] M. Hansen. Approximation algorithms for geometric embeddings in the plane with applications to parallel processing problems. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 604–609, October 1989.

[5] P. Klein, S. Plotkin, and S. Rao. Excluded minors, network decomposition and multicommodity flow. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, pages 682–690, October 1993.

[6] F. T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 422–431. IEEE Computer Society Press, October 1988.

[7] G. Ramalingam and C. Pandu Rangan. A unified approach to domination problems in interval graphs. *Information Processing Letters*, 27:271–174, 1988.

[8] R. Ravi, A. Agrawal, and P. Klein. Ordering problems approximated: single processor scheduling and interval graph completion. In *Proceedings of the 18th International Colloquium on Automata, Languages and Programming*, pages 751–762, July 1991.

[9] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.